Introduction to RBM package

Dongmei Li

April 24, 2017

Clinical and Translational Science Institute, University of Rochester School of Medicine and Dentistry, Rochester, NY 14642-0708

Contents

1 Overview

This document provides an introduction to the `RBM` package. The `RBM` package executes the resampling-based empirical Bayes approach using either permutation or bootstrap tests based on moderated t-statistics through the following steps.

- Firstly, the RBM package computes the moderated t-statistics based on the observed data set for each feature using the lmFit and eBayes function.
- Secondly, the original data are permuted or bootstrapped in a way that matches the null hypothesis to generate permuted or bootstrapped resamples, and the reference distribution is constructed using the resampled moderated t-statistics calculated from permutation or bootstrap resamples.
- Finally, the p-values from permutation or bootstrap tests are calculated based on the proportion of the permuted or bootstrapped moderated t-statistics that are as extreme as, or more extreme than, the observed moderated t-statistics.

Additional detailed information regarding resampling-based empirical Bayes approach can be found elsewhere (Li et al., 2013).

2 Getting started

The RBM package can be installed and loaded through the following R code.
Install the RBM package with:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("RBM")
```

Load the RBM package with:

```
> library(RBM)
```

3 RBM_T and RBM_F functions

There are two functions in the RBM package: RBM_T and RBM_F. Both functions require input data in the matrix format with rows denoting features and columns denoting samples. RBM_T is used for two-group comparisons such as study designs with a treatment group and a control group. RBM_F can be used for more complex study designs such as more than two groups or time-course studies. Both functions need a vector for group notation, i.e., "1" denotes the treatment group and "0" denotes the control group. For the RBM_F function, a contrast vector need to be provided by users to perform pairwise comparisons between groups. For example, if the design has three groups (0, 1, 2), the aContrast parameter will be a vector such as ("X1-X0", "X2-X1", "X2-X0") to denote all pairwise comparisons. Users just need to add an extra "X" before the group labels to do the contrasts.

- Examples using the RBM_T function: normdata simulates a standardized gene expression data and unifdata simulates a methylation microarray data. The p-values from the RBM_T function could be further adjusted using the p.adjust function in the stats package through the Bejamini-Hochberg method.

```
> library(RBM)
> normdata <- matrix(rnorm(1000*6, 0, 1),1000,6)
> mydesign <- c(0,0,0,1,1,1)
> myresult <- RBM_T(normdata,mydesign,100,0.05)
> summary(myresult)

              Length Class  Mode
ordfit_t      1000   -none- numeric
ordfit_pvalue 1000   -none- numeric
ordfit_beta0  1000   -none- numeric
ordfit_beta1  1000   -none- numeric
permutation_p 1000   -none- numeric
bootstrap_p   1000   -none- numeric

> sum(myresult$permutation_p<=0.05)

[1] 31
```

```
> which(myresult$permutation_p<=0.05)

 [1]  10  44  89 114 151 161 177 263 300 361 431 504 557 572 612 613 616 617 625
[20] 639 687 695 716 718 736 737 762 786 829 980 988

> sum(myresult$bootstrap_p<=0.05)

[1] 20

> which(myresult$bootstrap_p<=0.05)

 [1]   2  21  60  66  97 114 177 209 289 378 475 561 578 616 696 764 840 867 966
[20] 971

> permutation_adjp <- p.adjust(myresult$permutation_p, "BH")
> sum(permutation_adjp<=0.05)

[1] 0

> bootstrap_adjp <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adjp<=0.05)

[1] 0

> unifdata <- matrix(runif(1000*7,0.10, 0.95), 1000, 7)
> mydesign2 <- c(0,0,0, 1,1,1,1)
> myresult2 <- RBM_T(unifdata,mydesign2,100,0.05)
> sum(myresult2$permutatioin_p<=0.05)

[1] 0

> sum(myresult2$bootstrap_p<=0.05)

[1] 26

> which(myresult2$bootstrap_p<=0.05)

 [1]   6  51  64  96 114 128 173 226 288 300 303 333 421 424 449 515 660 673 675
[20] 679 707 727 841 870 911 979

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=0.05)

[1] 0
```

- Examples using the `RBM_F` function: normdata_F simulates a standardized gene expression data and unifdata_F simulates a methylation microarray data. In both examples, we were interested in pairwise comparisons.

```
> normdata_F <- matrix(rnorm(1000*9,0,2), 1000, 9)
> mydesign_F <- c(0, 0, 0, 1, 1, 1, 2, 2, 2)
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult_F <- RBM_F(normdata_F, mydesign_F, aContrast, 100, 0.05)
> summary(myresult_F)

              Length Class  Mode
ordfit_t      3000   -none- numeric
ordfit_pvalue 3000   -none- numeric
ordfit_beta1  3000   -none- numeric
permutation_p 3000   -none- numeric
bootstrap_p   3000   -none- numeric

> sum(myresult_F$permutation_p[, 1]<=0.05)

[1] 67

> sum(myresult_F$permutation_p[, 2]<=0.05)

[1] 73

> sum(myresult_F$permutation_p[, 3]<=0.05)

[1] 87

> which(myresult_F$permutation_p[, 1]<=0.05)

 [1]   8  14  54  83 105 114 130 131 157 164 189 201 206 228 232 233 250 261 279
[20] 303 316 339 354 387 388 398 406 407 416 434 483 495 500 527 557 570 571 572
[39] 577 621 634 648 650 657 661 674 682 691 692 709 743 747 750 784 786 792 805
[58] 810 819 827 867 895 901 954 956 963 968

> which(myresult_F$permutation_p[, 2]<=0.05)

 [1]   8  14  54  62 130 149 157 164 189 201 206 228 232 250 261 279 303 327 339
[20] 354 387 388 398 406 407 416 427 444 483 495 500 520 527 536 542 557 570 571
[39] 572 577 621 631 634 639 648 650 657 661 674 691 692 693 709 728 739 750 774
[58] 784 786 792 796 810 814 819 827 867 869 895 921 954 956 963 968

> which(myresult_F$permutation_p[, 3]<=0.05)

 [1]   8  14  54  62  79  83 105 114 130 131 157 164 189 201 206 228 232 233 250
[20] 261 279 297 303 316 327 328 339 345 348 354 363 387 388 398 406 407 427 434
[39] 444 447 483 495 500 520 527 536 550 557 570 571 572 573 575 577 621 634 639
[58] 648 650 657 661 674 691 692 693 709 728 750 773 784 786 792 796 810 814 819
[77] 827 867 869 895 920 921 956 963 968 983 997
```

```
> con1_adjp <- p.adjust(myresult_F$permutation_p[, 1], "BH")
> sum(con1_adjp<=0.05/3)

[1] 16

> con2_adjp <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adjp<=0.05/3)

[1] 9

> con3_adjp <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adjp<=0.05/3)

[1] 21

> which(con2_adjp<=0.05/3)

[1] 201 228 388 406 571 572 621 956 968

> which(con3_adjp<=0.05/3)

 [1]   8 130 164 201 228 232 279 339 387 388 406 571 572 621 674 750 792 810 814
[20] 956 968

> unifdata_F <- matrix(runif(1000*18, 0.15, 0.98), 1000, 18)
> mydesign2_F <- c(rep(0, 6), rep(1, 6), rep(2, 6))
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult2_F <- RBM_F(unifdata_F, mydesign2_F, aContrast, 100, 0.05)
> summary(myresult2_F)

             Length Class  Mode
ordfit_t     3000   -none- numeric
ordfit_pvalue 3000   -none- numeric
ordfit_beta1  3000   -none- numeric
permutation_p 3000   -none- numeric
bootstrap_p   3000   -none- numeric

> sum(myresult2_F$bootstrap_p[, 1]<=0.05)

[1] 44

> sum(myresult2_F$bootstrap_p[, 2]<=0.05)

[1] 44

> sum(myresult2_F$bootstrap_p[, 3]<=0.05)

[1] 52
```

```
> which(myresult2_F$bootstrap_p[, 1]<=0.05)

 [1]   1  20  37  69  77 103 150 158 197 202 207 225 228 237 256 258 272 365 403
[20] 424 425 457 459 460 491 500 516 548 619 630 633 661 688 698 711 727 752 766
[39] 788 826 855 878 908 964

> which(myresult2_F$bootstrap_p[, 2]<=0.05)

 [1]   1  20  69 103 150 158 197 202 203 207 228 232 256 258 288 365 403 424 425
[20] 437 457 459 460 500 516 619 630 678 682 688 711 724 727 752 766 788 826 831
[39] 844 855 878 908 964 978

> which(myresult2_F$bootstrap_p[, 3]<=0.05)

 [1]   1  20  69 103 125 150 158 197 202 207 225 228 256 258 272 288 341 365 403
[20] 424 425 457 459 460 491 500 516 585 619 630 639 652 678 682 688 693 698 711
[39] 719 724 727 752 766 788 811 826 831 834 855 878 908 964

> con21_adjp <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adjp<=0.05/3)

[1] 4

> con22_adjp <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adjp<=0.05/3)

[1] 5

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)

[1] 11
```

4 Ovarian cancer methylation example using the RBM_T function

Two-group comparisons are the most common contrast in biological and biomedical field. The ovarian cancer methylation example is used to illustrate the application of RBM_T in identifying differentially methylated loci. The ovarian cancer methylation example is taken from the gemone-wide DNA methylation profiling of United Kingdom Ovarian Cancer Population Study (UKOPS). This study used Illumina Infinium 27k Human DNA methylation Beadchip v1.2 to obtain DNA methylation profiles on over 27,000 CpGs in whole blood cells from 266 ovarian cancer women and 274 age-matched healthy controls. The data are downloaded from the NCBI GEO website with access number GSE19711. For illutration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovariance cancer cases (pre-treatment) and 4 healthy controls. The following codes show the process of generating significant differential DNA methylation loci using the RBM_T function and presenting the results for further validation and investigations.

```
> system.file("data", package = "RBM")

[1] "/tmp/RtmpgnSTFp/Rinst706e4bc76c12/RBM/data"

> data(ovarian_cancer_methylation)
> summary(ovarian_cancer_methylation)

        IlmnID         Beta          exmdata2[, 2]     exmdata3[, 2]
 cg00000292:  1   Min.   :0.01058   Min.   :0.01187   Min.   :0.009103
 cg00002426:  1   1st Qu.:0.04111   1st Qu.:0.04407   1st Qu.:0.041543
 cg00003994:  1   Median :0.08284   Median :0.09531   Median :0.087042
 cg00005847:  1   Mean   :0.27397   Mean   :0.28872   Mean   :0.283729
 cg00006414:  1   3rd Qu.:0.52135   3rd Qu.:0.59032   3rd Qu.:0.558575
 cg00007981:  1   Max.   :0.97069   Max.   :0.96937   Max.   :0.970155
 (Other)   :994                     NA's   :4
 exmdata4[, 2]     exmdata5[, 2]     exmdata6[, 2]     exmdata7[, 2]
 Min.   :0.01019   Min.   :0.01108   Min.   :0.01937   Min.   :0.01278
 1st Qu.:0.04092   1st Qu.:0.04059   1st Qu.:0.05060   1st Qu.:0.04260
 Median :0.09042   Median :0.08527   Median :0.09502   Median :0.09362
 Mean   :0.28508   Mean   :0.28482   Mean   :0.27348   Mean   :0.27563
 3rd Qu.:0.57502   3rd Qu.:0.57300   3rd Qu.:0.52099   3rd Qu.:0.52240
 Max.   :0.96658   Max.   :0.97516   Max.   :0.96681   Max.   :0.95974
                   NA's   :1
 exmdata8[, 2]
 Min.   :0.01357
 1st Qu.:0.04387
 Median :0.09282
 Mean   :0.28679
 3rd Qu.:0.57217
 Max.   :0.96268

> ovarian_cancer_data <- ovarian_cancer_methylation[, -1]
> label <- c(1, 1, 0, 0, 1, 1, 0, 0)
> diff_results <- RBM_T(aData=ovarian_cancer_data, vec_trt=label, repetition=100, alpha=0.05)
> summary(diff_results)

             Length Class  Mode
ordfit_t     1000   -none- numeric
ordfit_pvalue 1000   -none- numeric
ordfit_beta0  1000   -none- numeric
ordfit_beta1  1000   -none- numeric
permutation_p 1000   -none- numeric
bootstrap_p   1000   -none- numeric

> sum(diff_results$ordfit_pvalue<=0.05)

[1] 45
```

```
> sum(diff_results$permutation_p<=0.05)

[1] 34

> sum(diff_results$bootstrap_p<=0.05)

[1] 56

> ordfit_adjp <- p.adjust(diff_results$ordfit_pvalue, "BH")
> sum(ordfit_adjp<=0.05)

[1] 0

> perm_adjp <- p.adjust(diff_results$permutation_p, "BH")
> sum(perm_adjp<=0.05)

[1] 0

> boot_adjp <- p.adjust(diff_results$bootstrap_p, "BH")
> sum(boot_adjp<=0.05)

[1] 2

> diff_list_perm <- which(perm_adjp<=0.05)
> diff_list_boot <- which(boot_adjp<=0.05)
> sig_results_perm <- cbind(ovarian_cancer_methylation[diff_list_perm, ], diff_results$ordfit_t
> print(sig_results_perm)

 [1] IlmnID
 [2] Beta
 [3] exmdata2[, 2]
 [4] exmdata3[, 2]
 [5] exmdata4[, 2]
 [6] exmdata5[, 2]
 [7] exmdata6[, 2]
 [8] exmdata7[, 2]
 [9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_perm]
[11] diff_results$permutation_p[diff_list_perm]
<0 rows> (or 0-length row.names)

> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t
> print(sig_results_boot)

        IlmnID       Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
285 cg00263760 0.09050395     0.1019776    0.14801710    0.12242400
882 cg00858899 0.11427700     0.1191954    0.07690343    0.08321229
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
```

```
285    0.11693600     0.1065043     0.1228116    0.12310430
882    0.08961409     0.1073066     0.0920398    0.08726349
    diff_results$ordfit_t[diff_list_boot]
285                              -3.093997
882                               3.179415
    diff_results$bootstrap_p[diff_list_boot]
285                                       0
882                                       0
```